CCS-3

Identifying and Eliminating the Performance Variability on ASCI Q

Fabrizio Petrini

Darren Kerbyson, Adolfy Hoisie, Scott Pakin, Harvey Wasserman, Juan Fernandez-Peinador, David Addison, Mike Lang

Performance and Architectures Laboratory (PAL)

CCS-3, Los Alamos National Laboratory

March 2003

LAUR 03-0528

Funded by: ASCI Ongoing Computing

Los Alamos

Talk Overview

CCS-3

- **Performance assessment of ASCI Q**
- **Our performance expectations for QA/QB, and the reality**
- **Identification of performance factors**
  - Application performance and breakdown into components
  - Application performance variability
  - Configuration of system
- **Detailed examination of system effects**
  - Identification of O/S effects
  - Effect of scaling – 1024 nodes and beyond
  - Quantification of the impact on achievable performance
- **Towards the Elimination of overheads**
  - Average improvement of 55% (peak 85%) on 3716 processors of the Q machine

Report on this work is available from
www.c3.lanl.gov/~fabrizio/publications.html

"Identifying and Eliminating the Performance Variability on ASCI Q", LA-UR-03-0138.

Los Alamos

CCS-3

Los Alamos

System Research

CCS-3

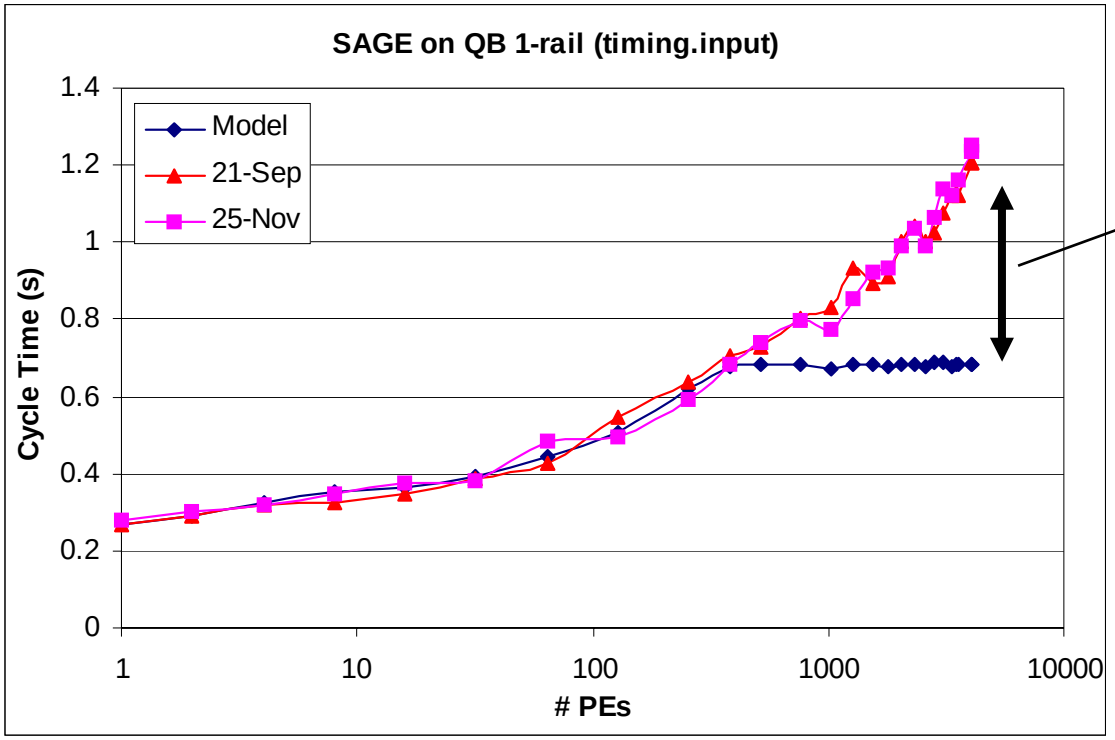
- **Scalable resource management (STORM): job launching on thousands of processors in few hundreds of milliseconds**
- **Job scheduling and coscheduling: increased throughput**
- **Performance evaluation of high performance networks**
- **System-level fault-tolerance**
- **Communication libraries: “deterministic” MPI**

Los Alamos

Modeled and Measured Performance of Sage

CCS-3

- **Predictions available from PAL performance Model**
- **Latest two sets of measurements are consistent (> 80% longer than model)**



There is a difference
why ?

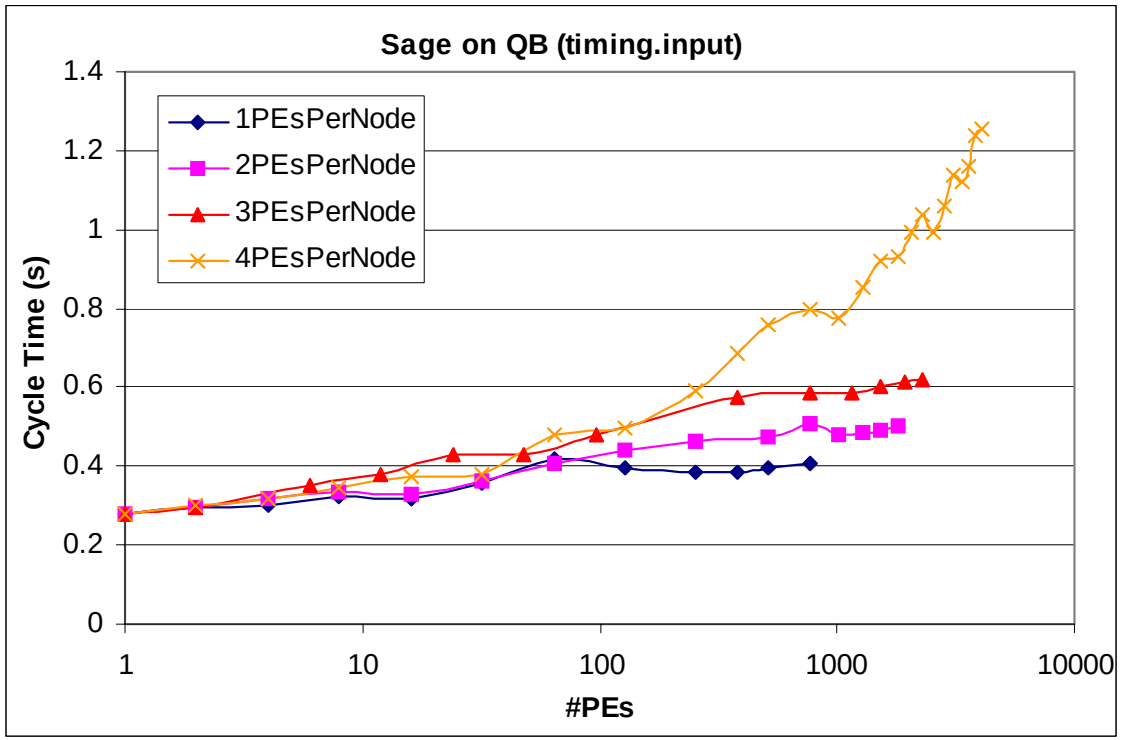
Lower is better!

Los Alamos

Using fewer PEs per Node

CCS-3

- **Test performance using 1,2,3 and 4 PEs per node**
- **N.B. reduces the number of compute processors available**

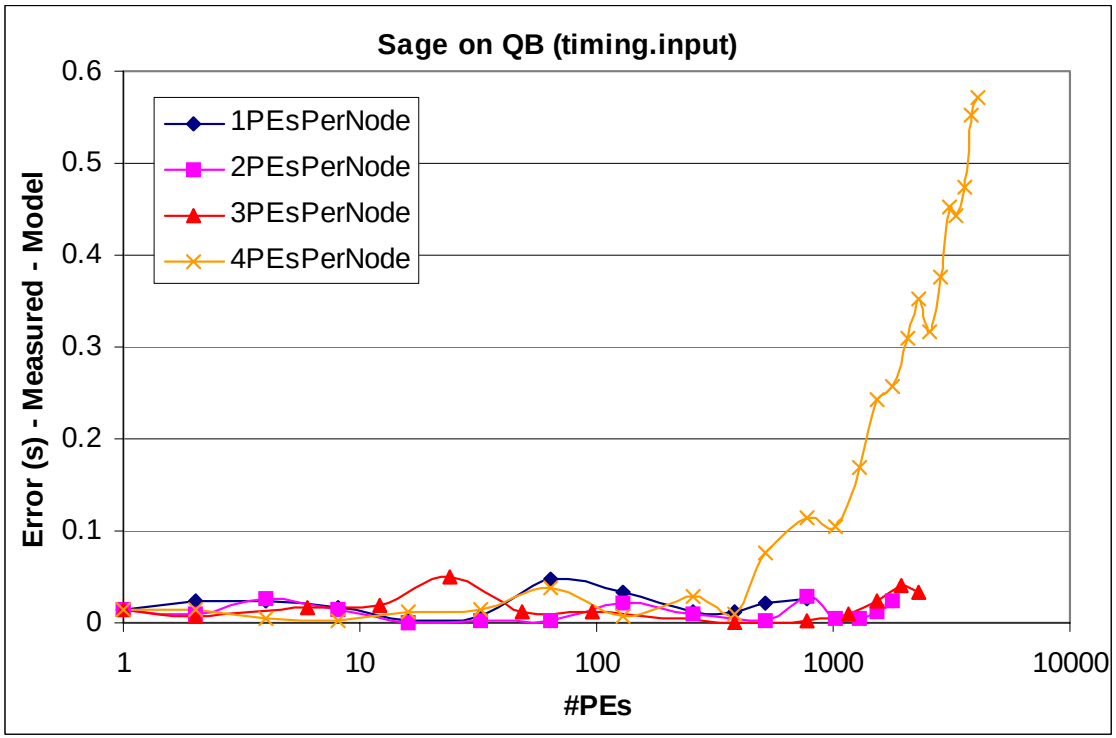


Los Alamos

Using fewer PEs per node (2)

CCS-3

- **Measurements match model almost exactly for 1,2 and 3 PEs per node!**



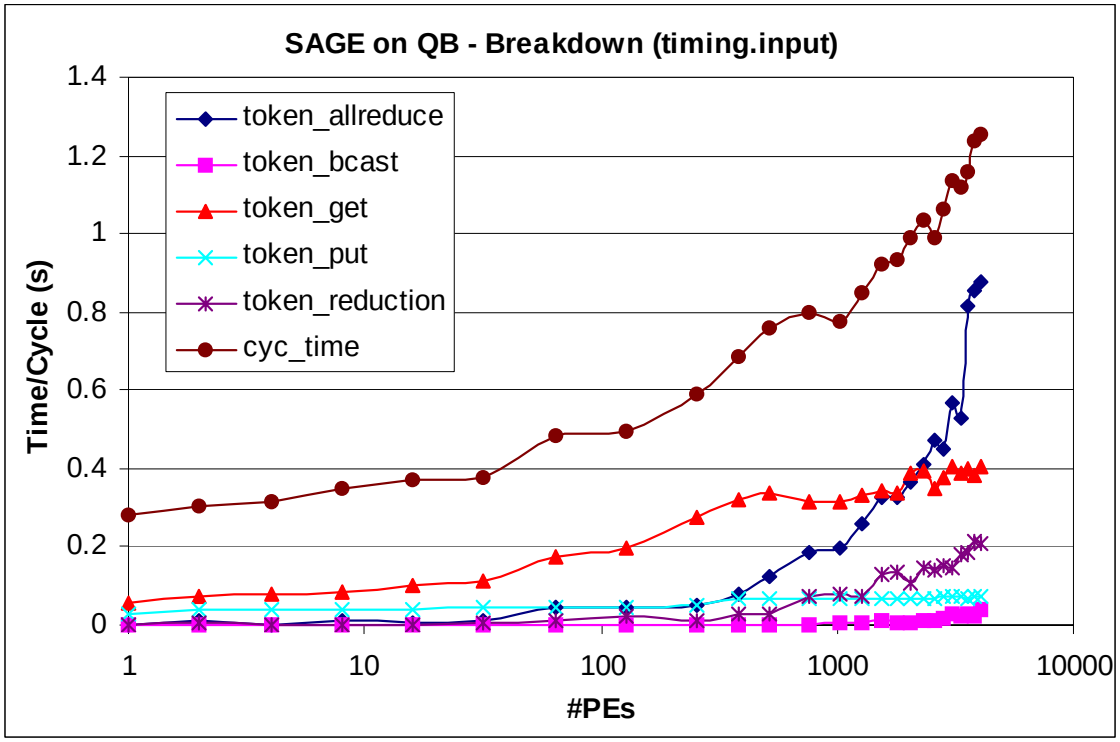
Performance issue only occurs when using 4 PEs per node

Los Alamos

SAGE performance components

CCS-3

- **Look at SAGE in terms of main components:**
  - Put/Get (point-to-point boundary exchange)
  - Collectives (allreduce, broadcast, reduction)



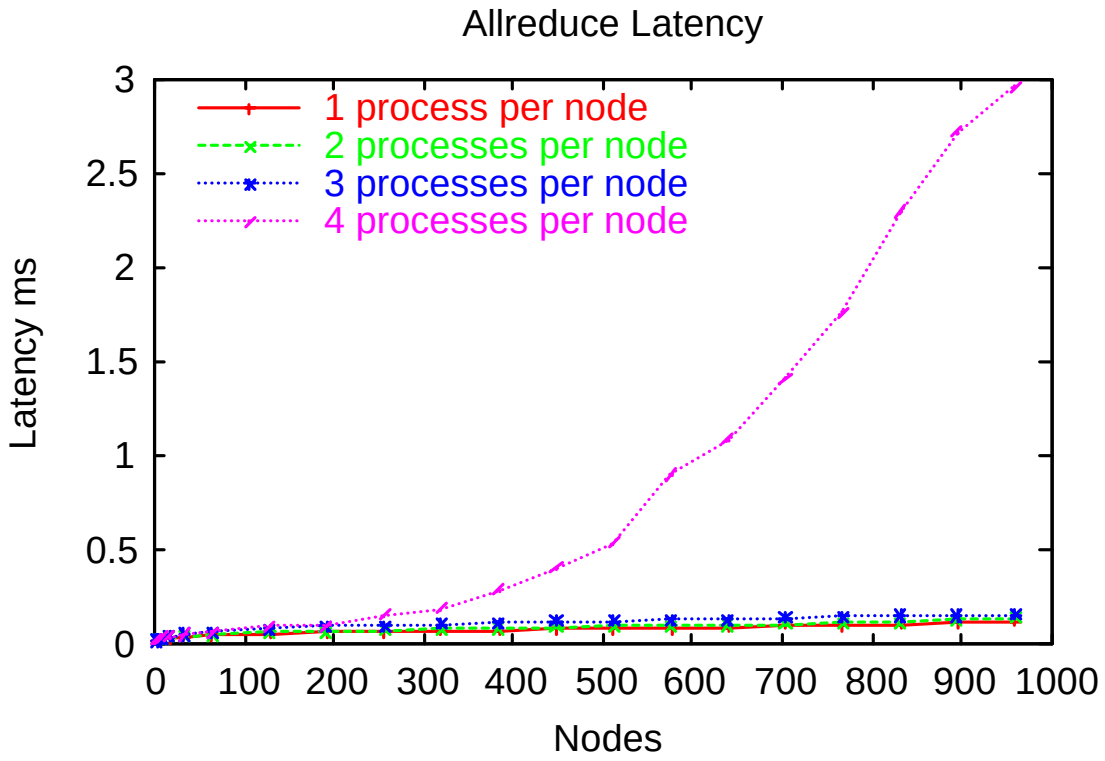
Performance issue seems to occur only on collective operations

Los Alamos

Performance of the collectives

CCS-3

- **Measure collective performance separately**

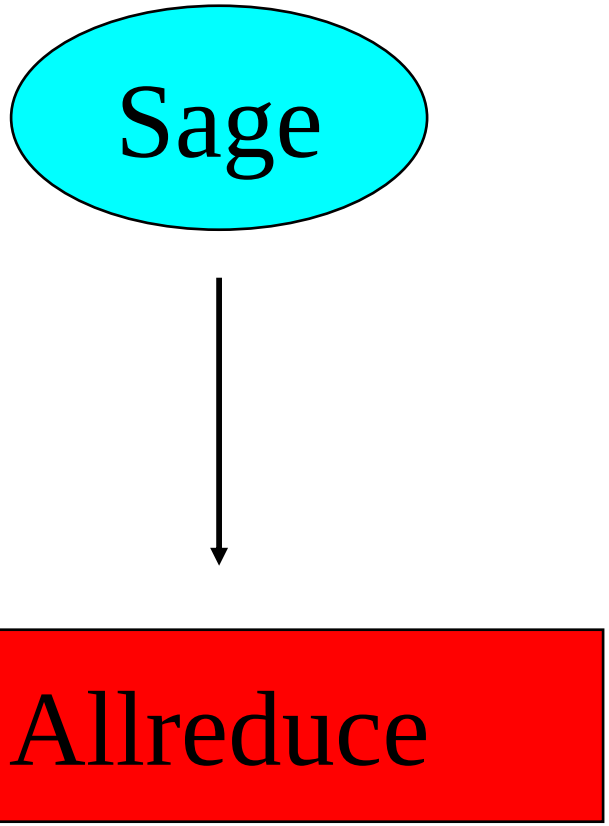


- **Collectives mirror the performance of the application**

Los Alamos

Identifying the problem within Sage

CCS-3



Simplify

Exposing the problems with simple benchmarks

CCS-3

Allreduce

Challenge: identify the simplest benchmark that exposes the problem

Benchmarks

Los Alamos

Interconnection network and communication libraries

CCS-3

- **The initial (obvious) suspects were the interconnection network and the MPI implementation**
- **We tested in depth the network, the low level transmission protocols and several allreduce algorithms**
- **We also implemented allreduce in the Network Card**
- **By changing the synchronization mechanism we were able to reduce the latency of an allreduce benchmark by a factor of 7**
- **But we only got small improvements in Sage (5%)**

We were not able to link the performance variability problems to either the network or MPI

Los Alamos

Computational noise

CCS-3

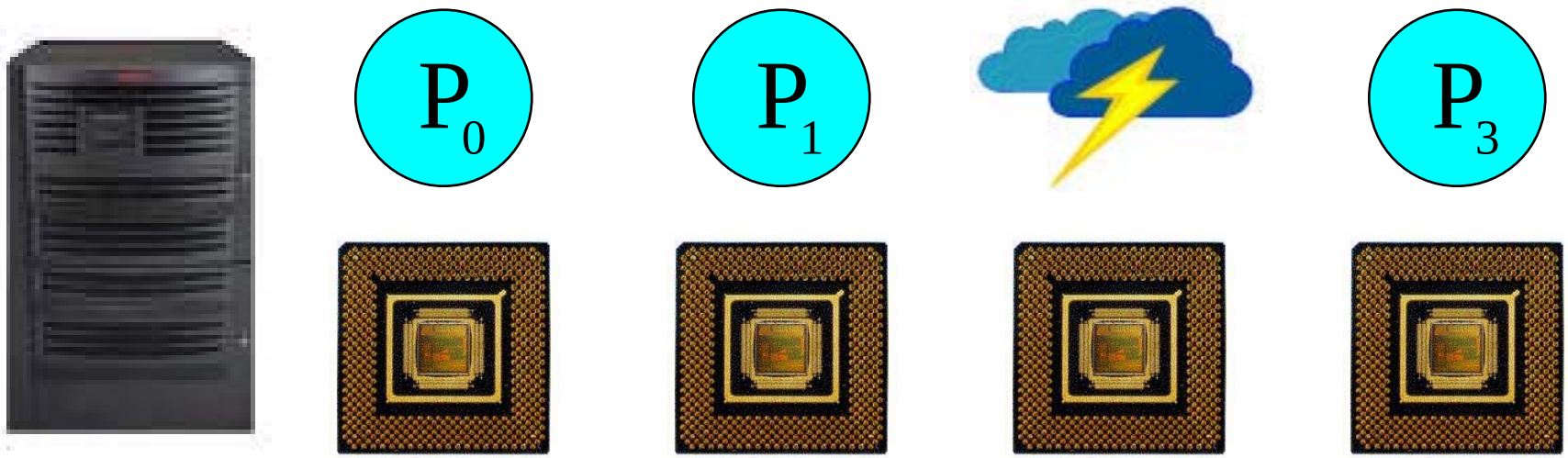
- **After having ruled out the network and MPI we focused our attention on the compute nodes**
- **Our hypothesis is that the noise is generated inside the processing nodes**
- **The noise “freezes” a running process for a certain amount of time and generates a “computational” hole**

Los Alamos

Computational noise: intuition

CCS-3

- **We are running 4 processes on 4 distinct processors on an Alphaserver ES45**
- **The computation of one process is interrupted by an external event (e.g., system daemon or kernel)**

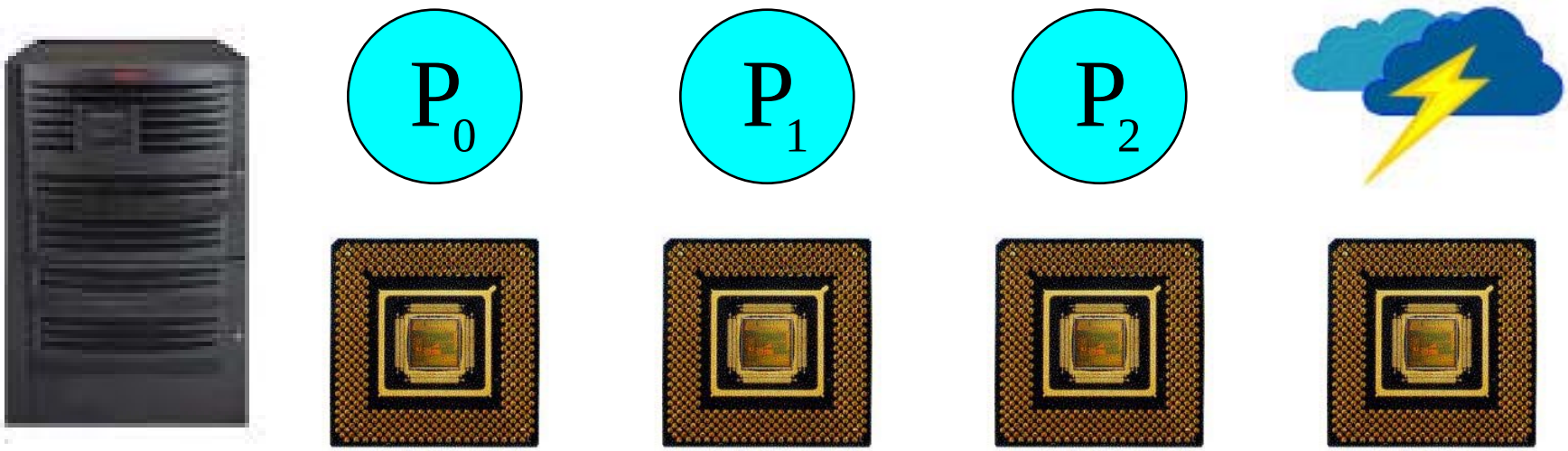


Los Alamos

Computational noise: 3 processes on 3 processors

CCS-3

- **We are running 3 processes on 3 distinct processors on Alphaserver ES45**
- **The “noise” can run on the 4th processor without interrupting the other 3 processes**



Los Alamos

A simple benchmark

CCS-3

- **In order to narrow the search space we implemented a simple benchmark that consumes CPU cycles and:**
  - does not perform any I/O
  - does not perform any communication
  - does not perform any synchronization
  - does not perform any main memory access
  - does not perform any cache access

- **We ran this benchmark in isolation on the whole QB (4096 processors/1024 nodes) after a global reboot**

Los Alamos

A simple benchmark

CCS-3

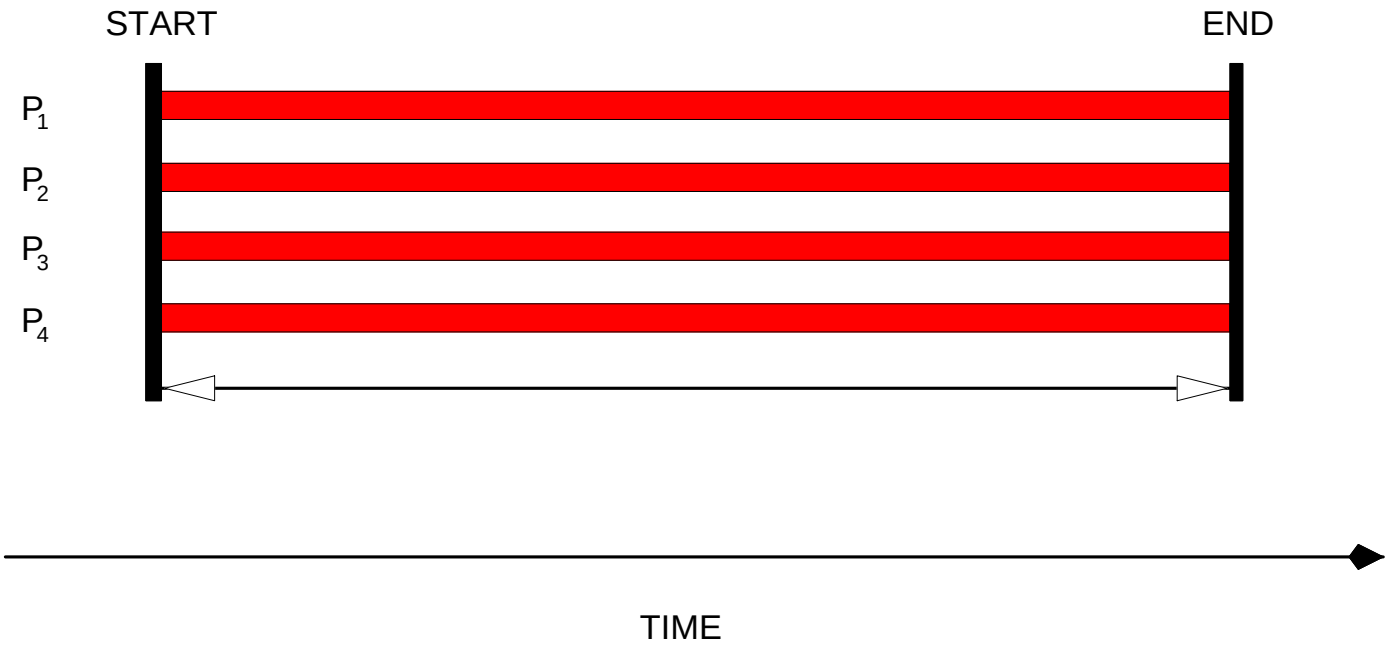
```
double compute( double work )
{
  unsigned_int64_t i, dum;
  timer_t t1, t2;
  double t;

  get_clock( t1 );

  /* waste CPU cycles in an amount approximate to the amount of work */
  for (i =  COMPUTE_K * work; i; i--)
    dum = i + 1;

  get_clock( t2 );
  diff( t, t2, t1 );
  return t;
}
```

Los Alamos

Coarse grained measurement

CCS-3

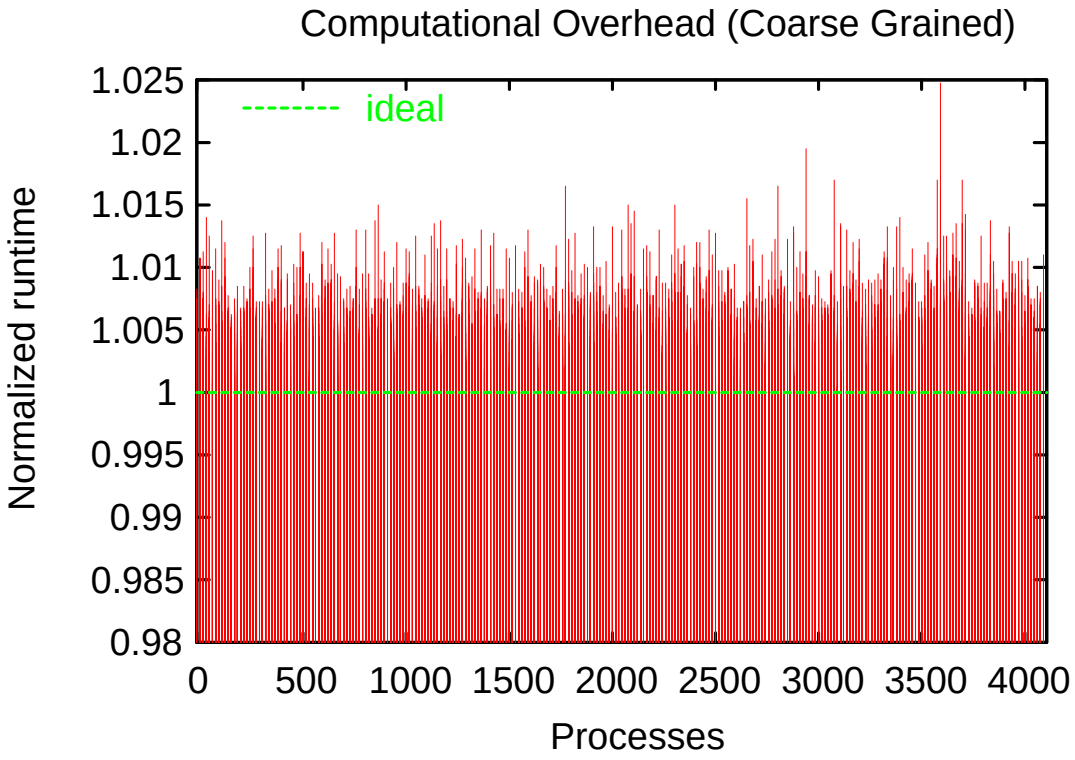
- **We execute the computational loop for 1000 seconds on all 4096 nodes of QB**



Los Alamos

Coarse grained computational overhead per process

- **The slowdown per process is small, between 1% and 2.5%**



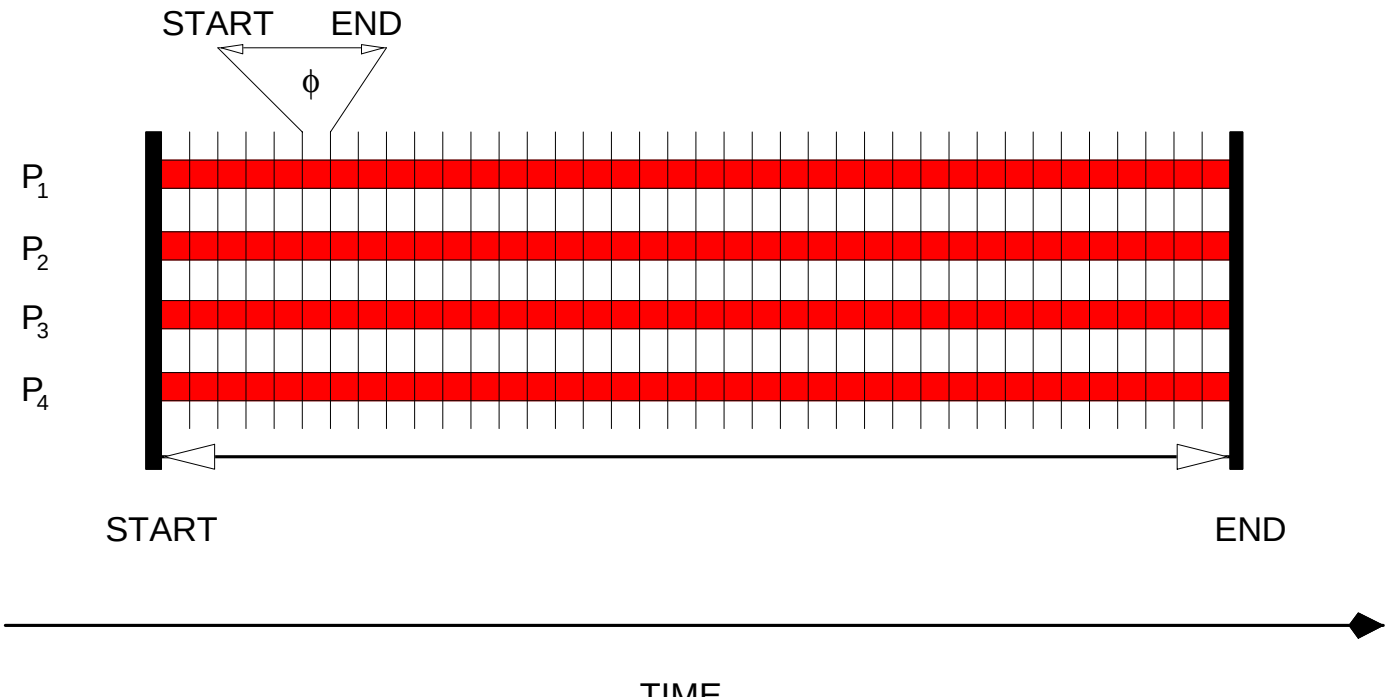
Optimum (lower is better)

Los Alamos

Fine grained measurement

CCS-3

- **We run the same benchmark for 1000 seconds, but we measure the run time every millisecond**
- **Fine granularity representative of many ASCI codes**

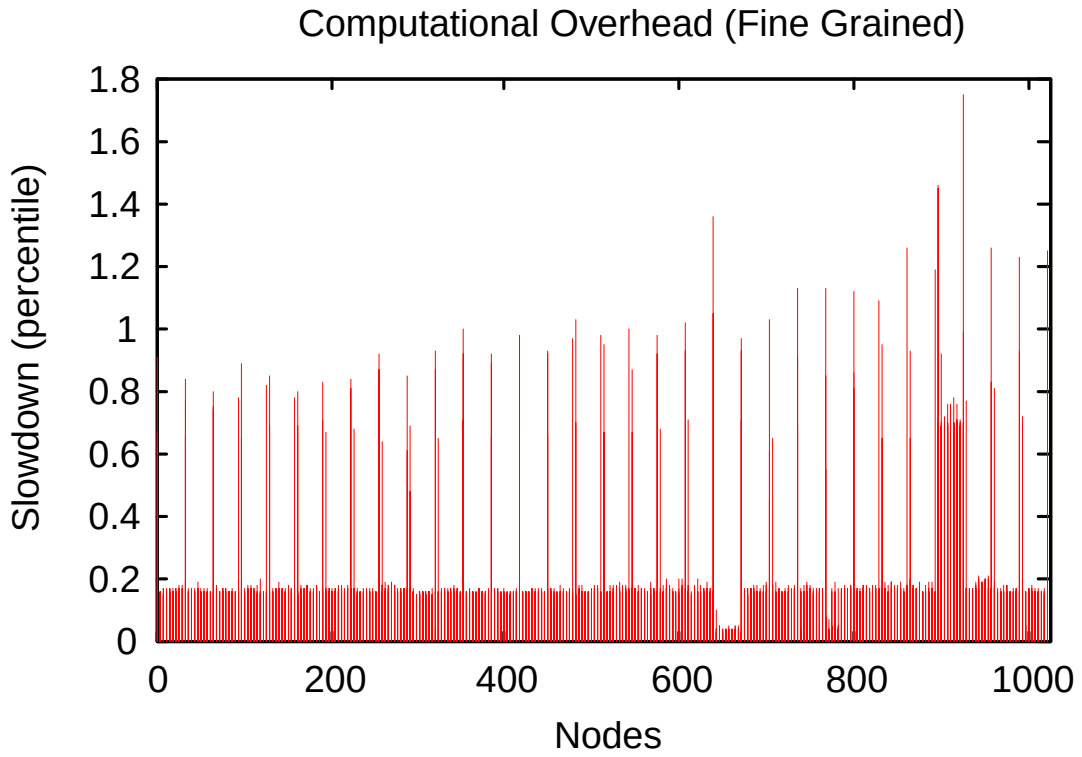


Los Alamos

Fine grained computational overhead per node

CCS-3

- **We now compute the slowdown per-node, rather than per-process**
- **The noise has a clear, per cluster, structure**



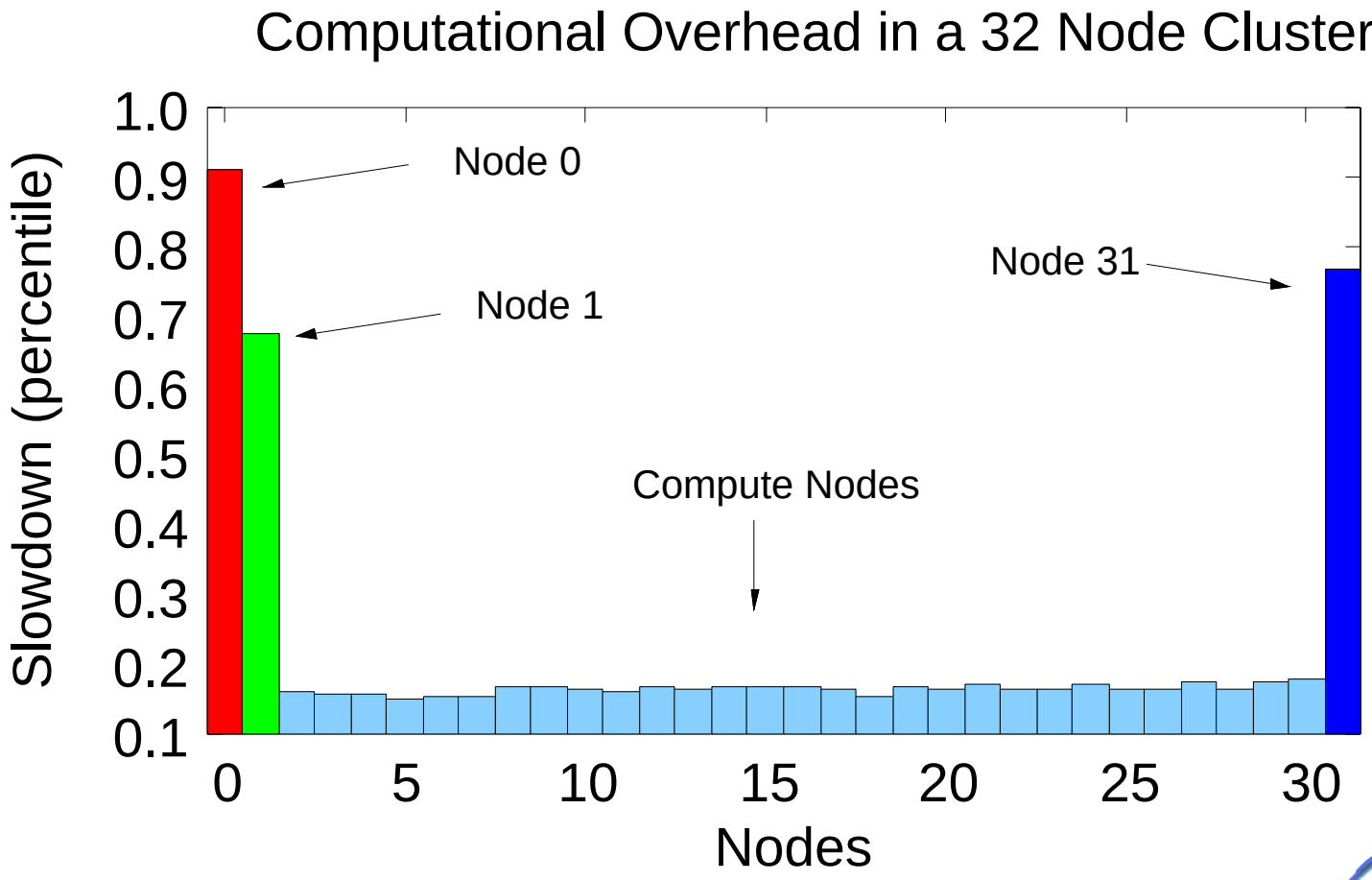
Optimum is 0
(lower is better)

Los Alamos

Noise in a 32 Node Cluster

CCS-3

- **The Q machine is organized in 32 node clusters (TruCluster)**
- **In each cluster there is a cluster manager (node 0), a quorum node (node 1) and the RMS data collection (node 31)**



Los Alamos

Per node noise distribution

CCS-3

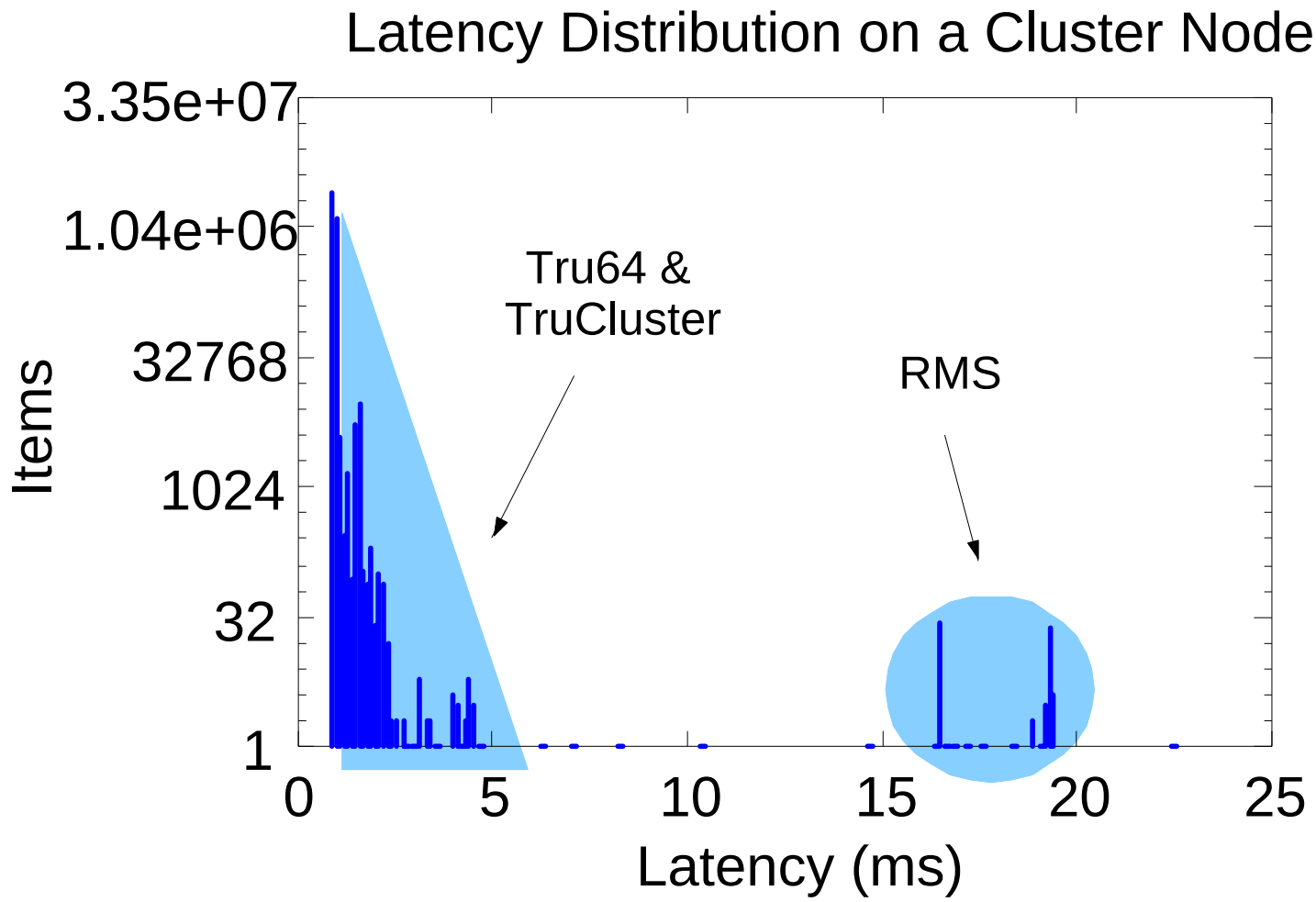
- **Plot distribution of one million, 1 ms computational chunks**
- **In an ideal, noiseless, machine the distribution graph is**
  - a single bar at 1 ms of 1 million points per process (4 million per node)
- **Every outlier identifies a computation that was delayed by external interference**
- **We show the distributions for the standard cluster node, and also nodes 0, 1 and 31**

Los Alamos

Cluster Node (2-30)

CCS-3

- **10% of the times the execution of the 1 ms chunk of computation is delayed**

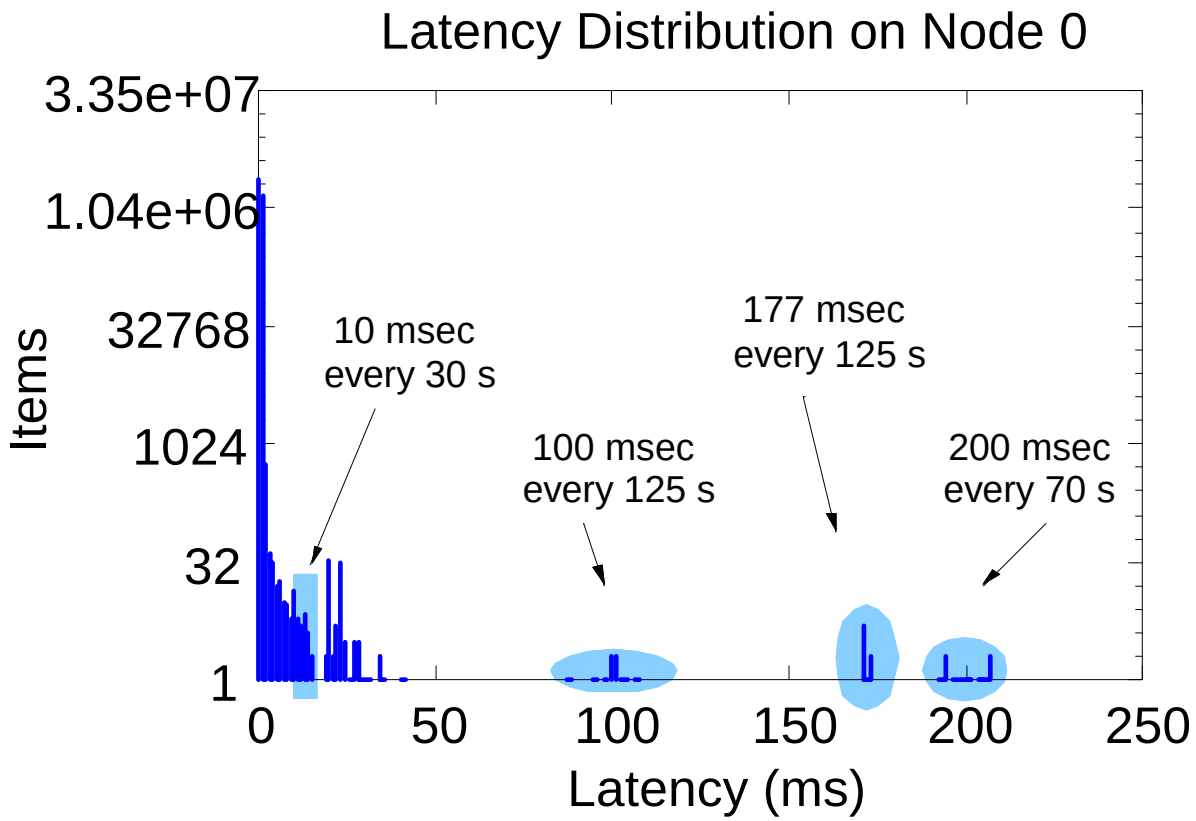


Los Alamos

Node 0, Cluster Manager

CCS-3

- **We can identify 4 main sources of noise**

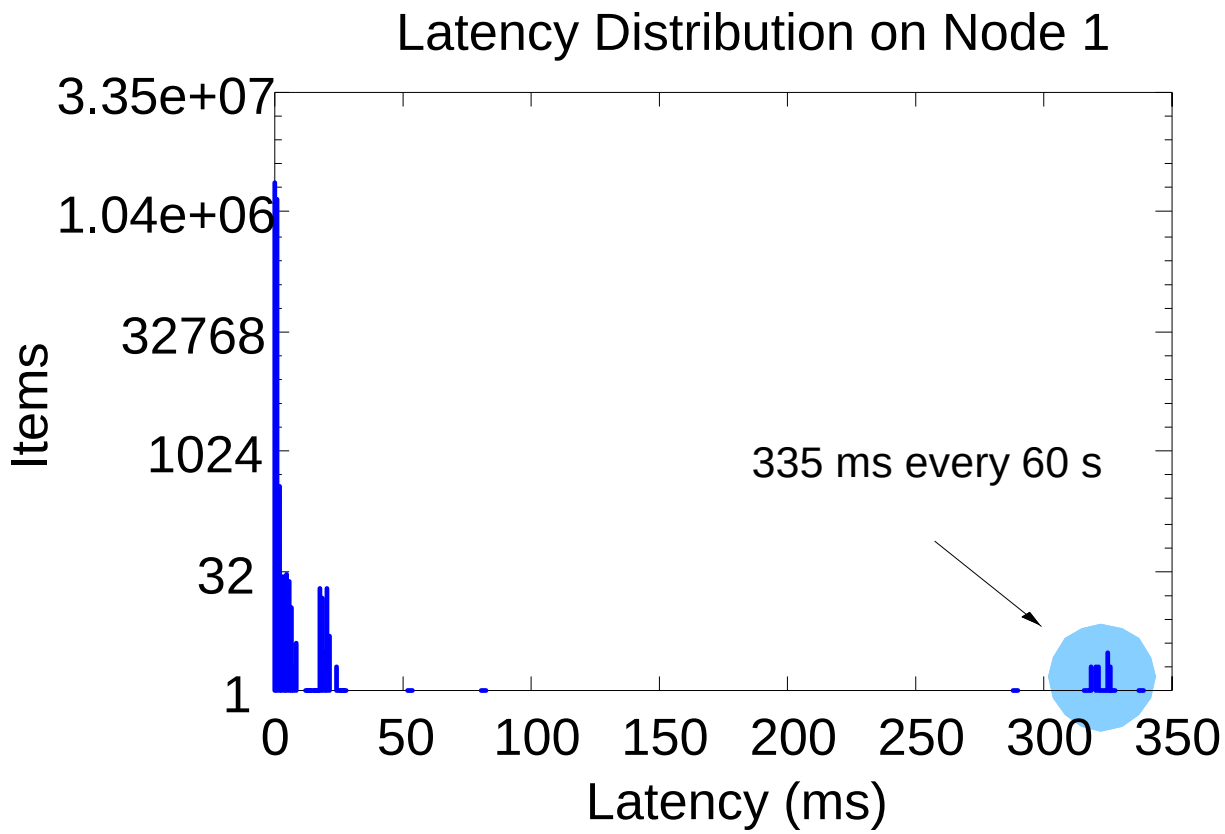


Los Alamos

Node 1, Quorum Node

CCS-3

- **One source of heavyweight noise (335 ms!)**

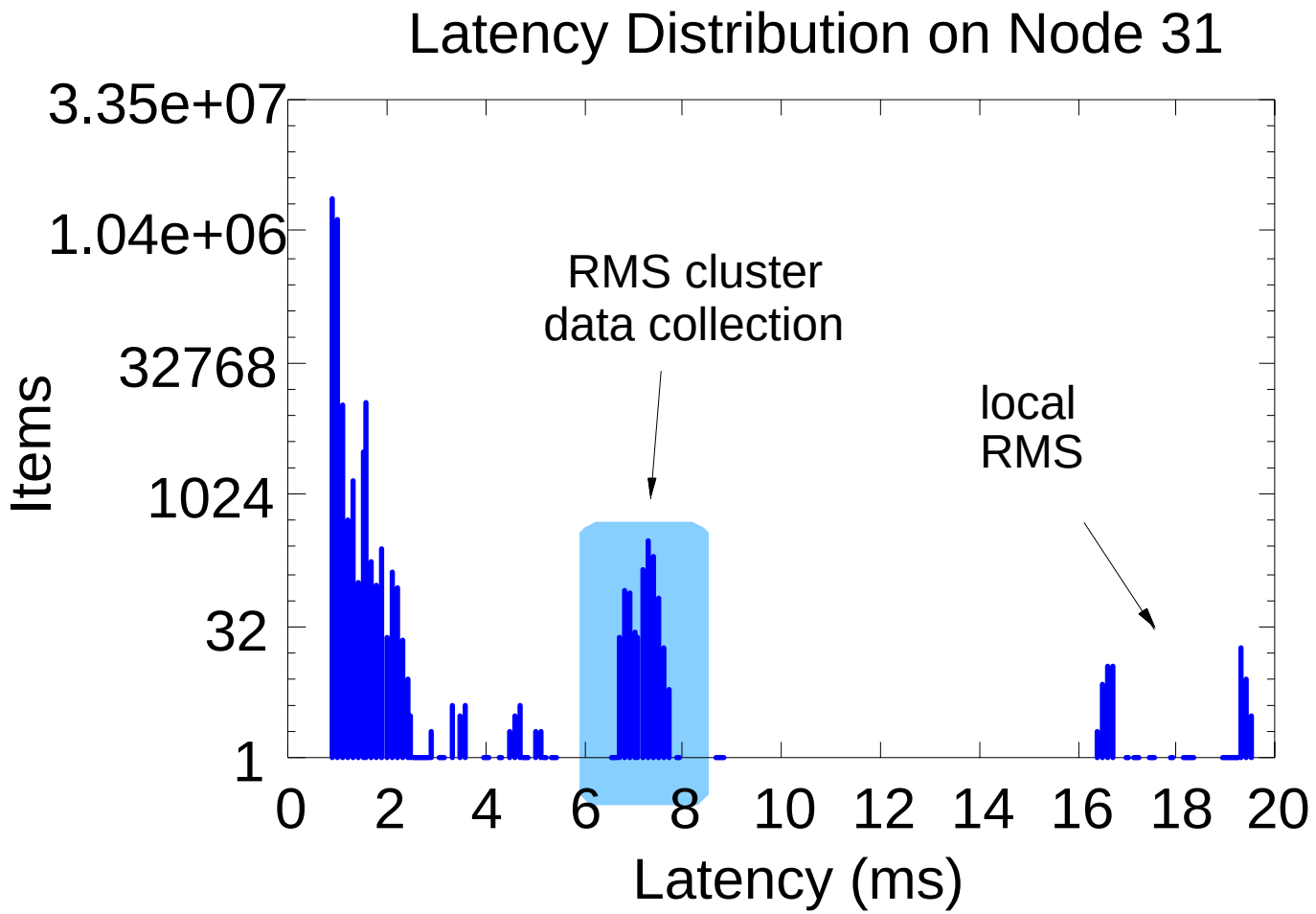


Los Alamos

Node 31

CCS-3

- **Many fine grained interruptions, between 6 and 8 milliseconds**

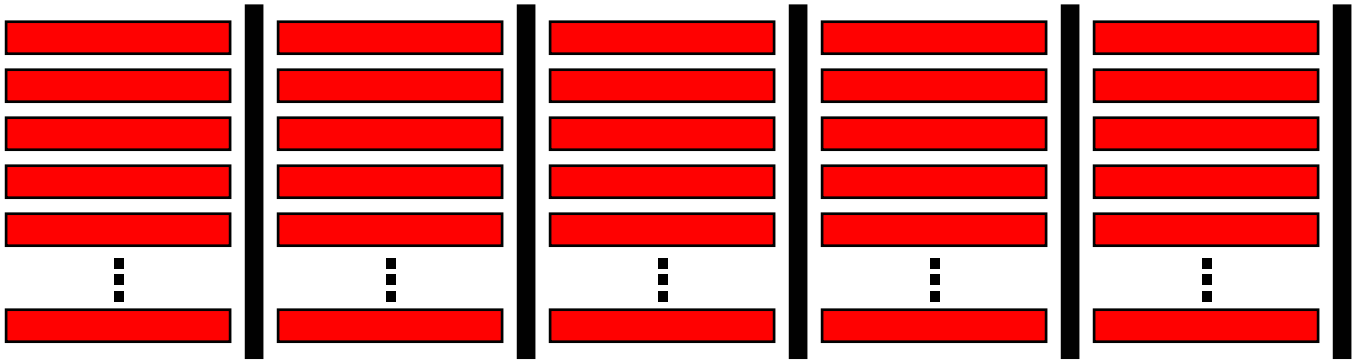


Los Alamos

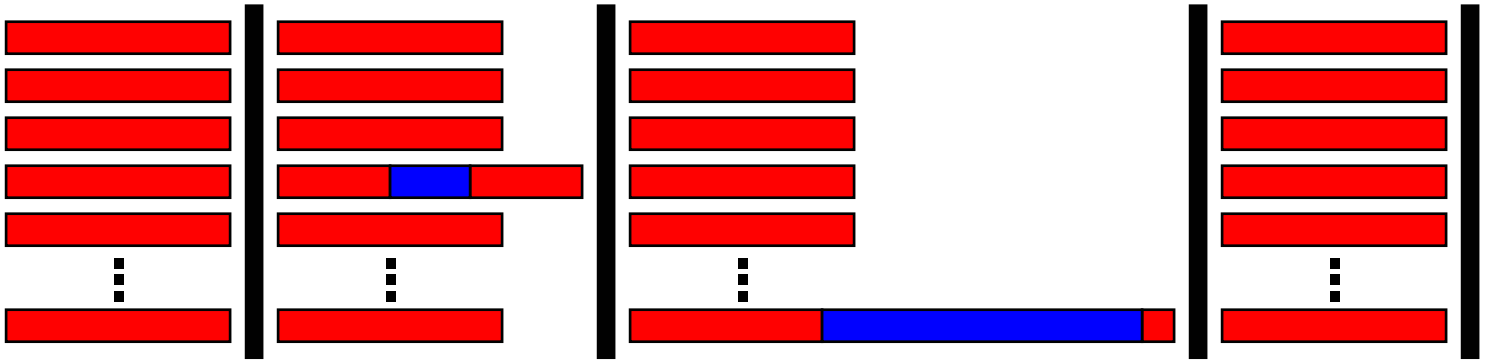
The effect of the noise

CCS-3

- **An application is usually a sequence of a computation followed by a synchronization (collective):**

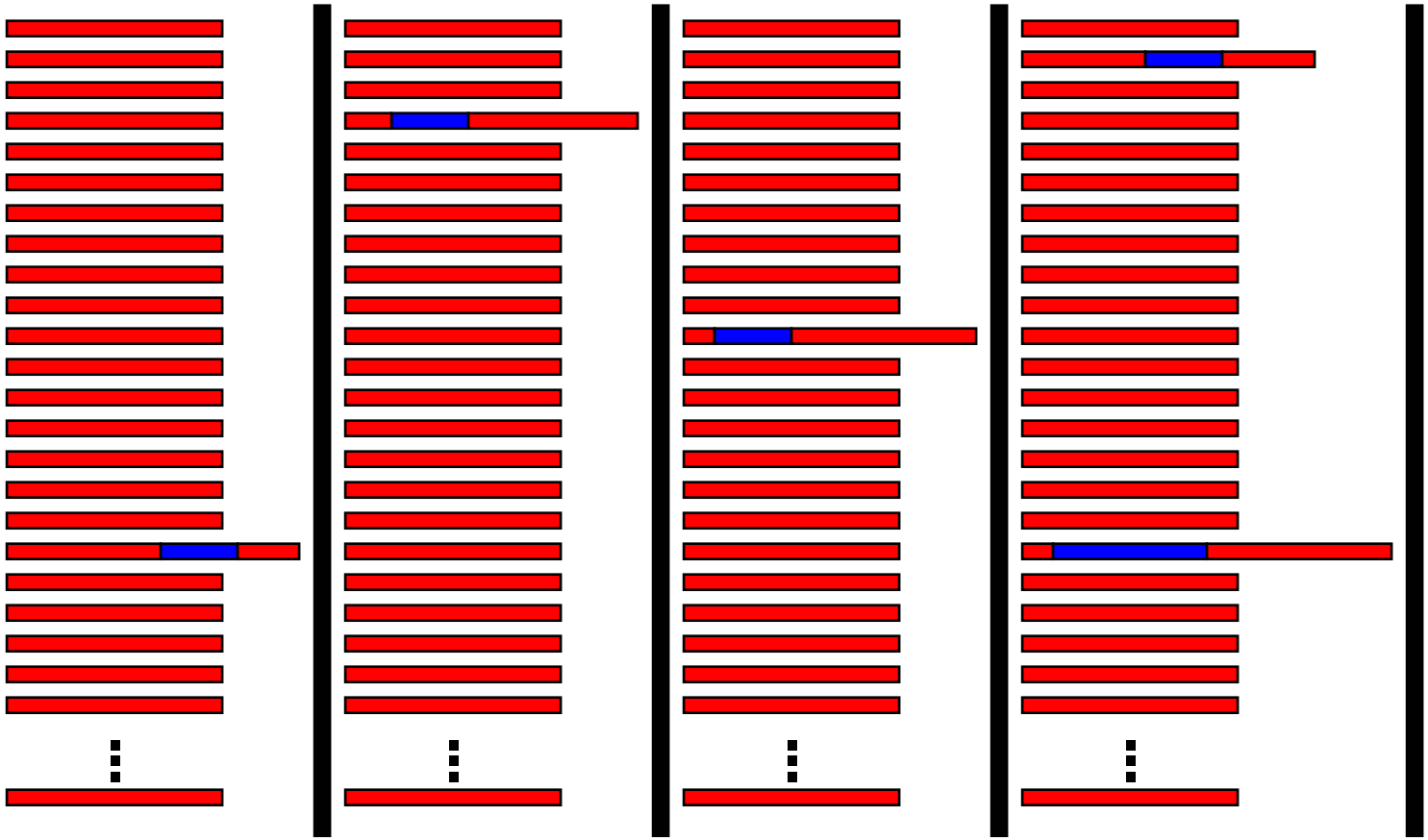


- **But if an event happens on a single node then it can affect all the other nodes**



Los Alamos

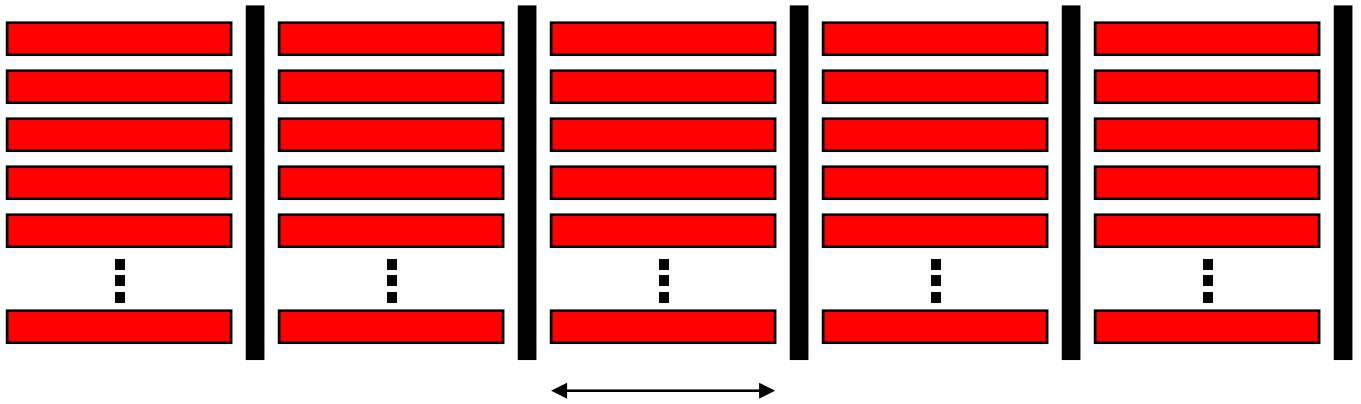
Effect of System Size



- **The probability of a random event occurring increases with the node count.**

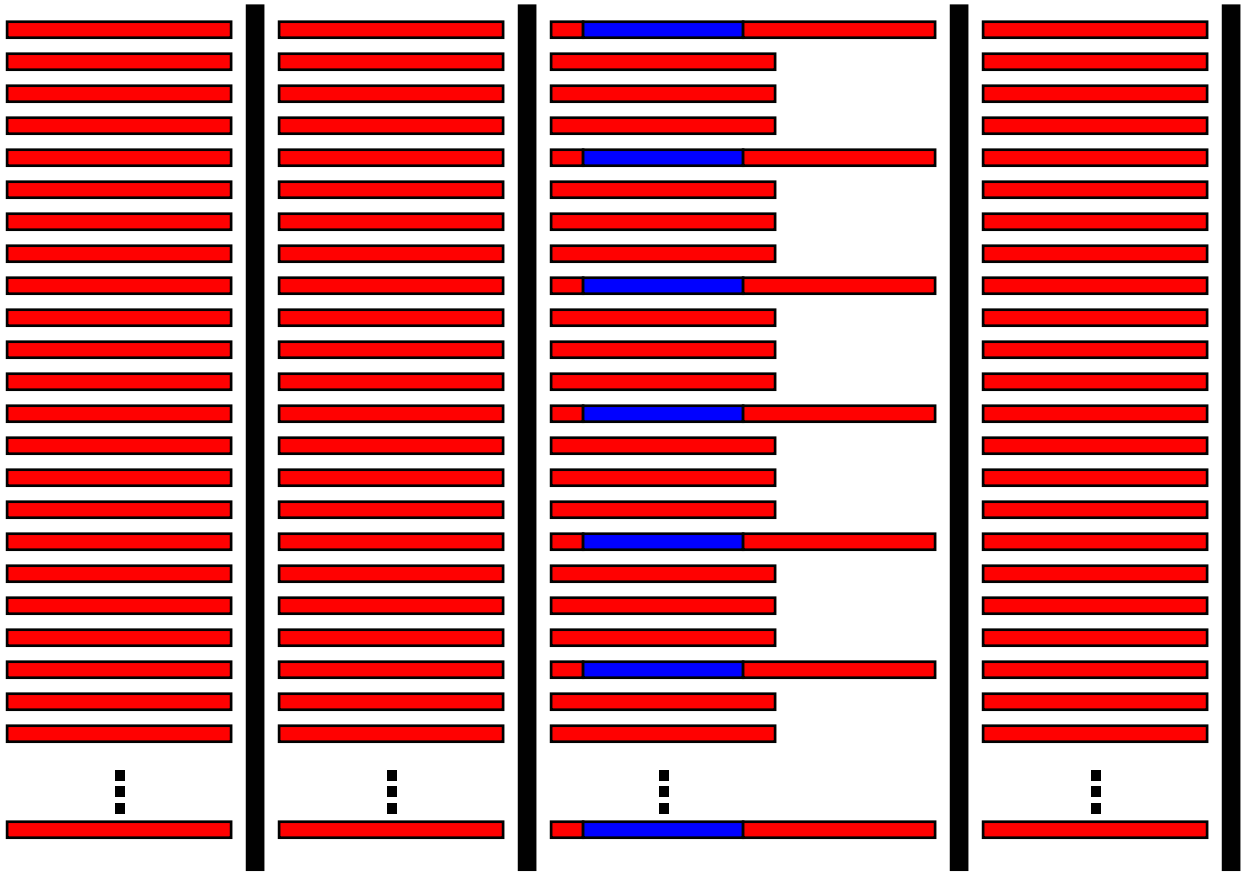
Noise and allreduce

- **We now try to correlate the computational noise in the nodes with the allreduce, by adding a global barrier at the end of each computational chunk of the basic benchmark**



Computational granularity

Effect of Co-scheduling

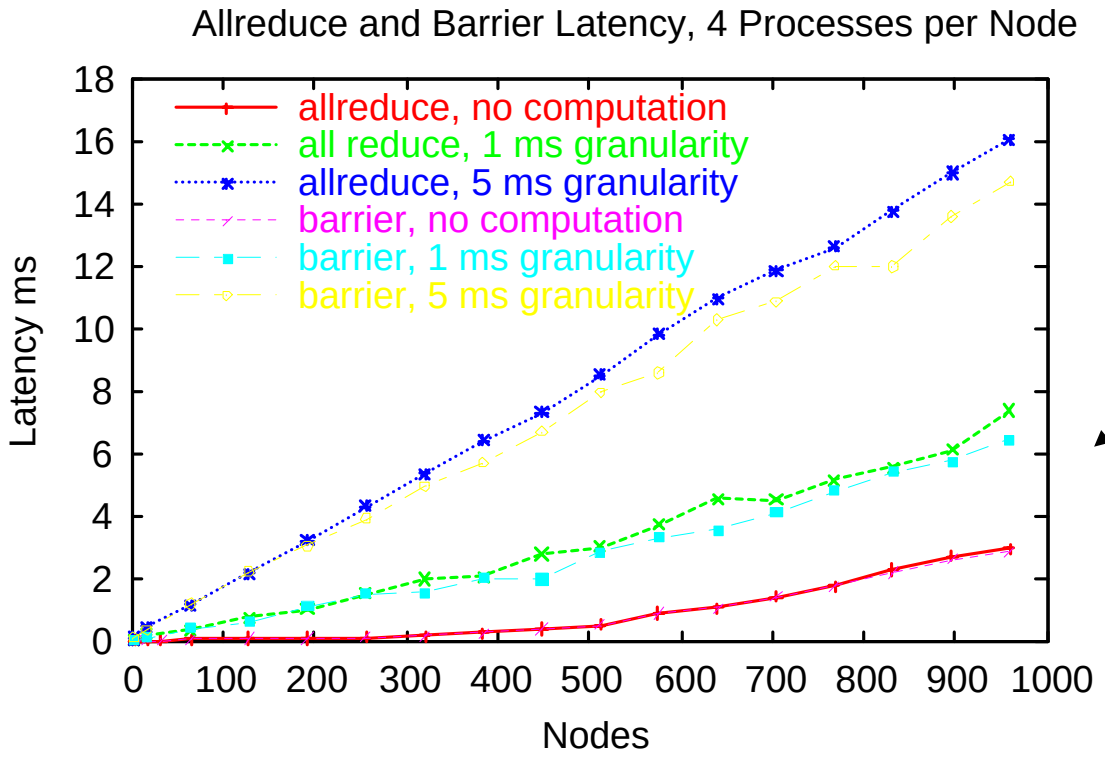


- **O/S events will occur at the same time on different nodes hence reducing impact on performance**

Barrier vs Allreduce

CCS-3

- **The results show that there is very little difference between barrier (executed in HW, which is almost instantaneous) and allreduce**
- **The graph confirms that the problem is in the process skew and not in the network**



Lower
Is better

Los Alamos

A discrete event simulator to analyze the noise

CCS-3

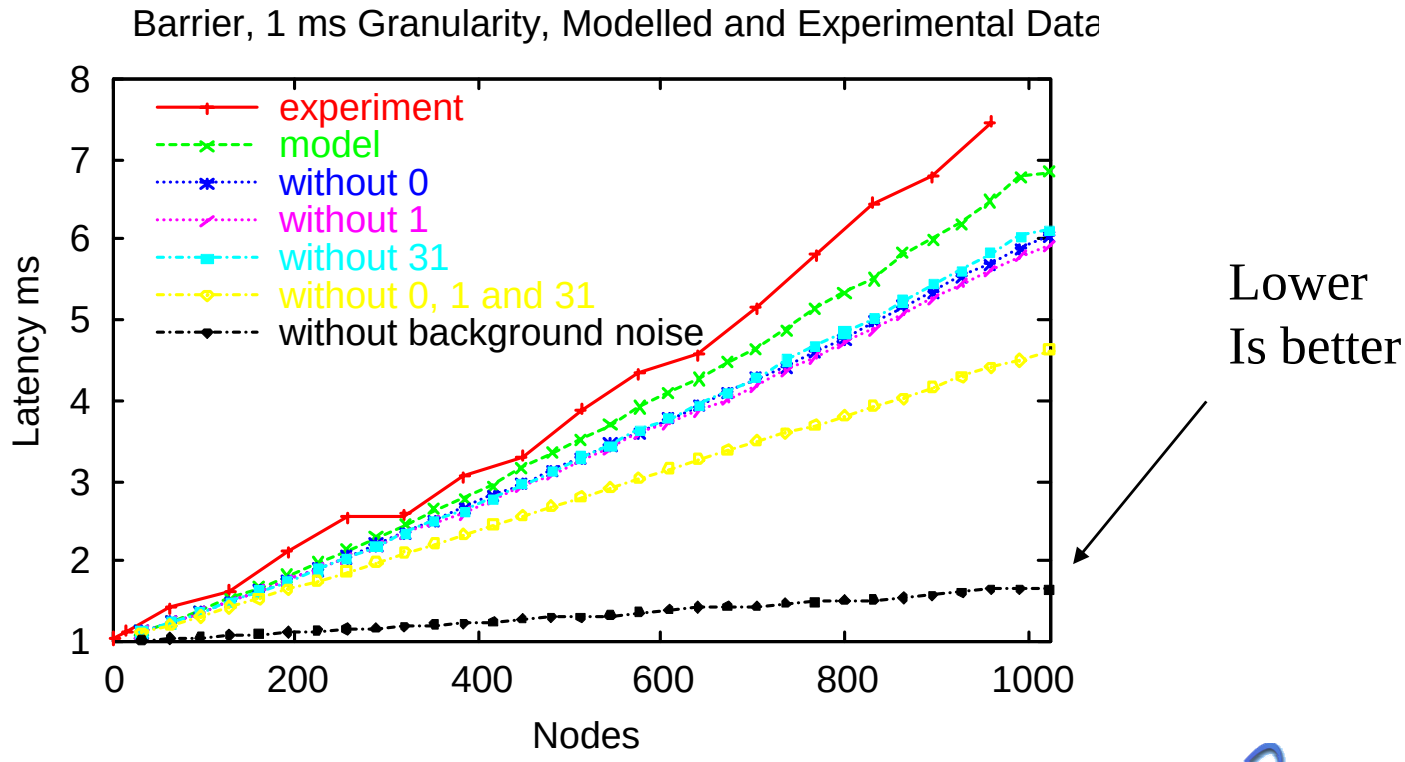
- **We developed a discrete event simulator to analyze the impact of each single source of noise**
- **With this simulator we can selectively remove sources of noise**
- **We can also explore the impact of the noise on larger configurations and different computational granularities without running the actual experiments**

What is the primary source of noise?

Los Alamos

Modeled and Experimental Data

- **The model is a close approximation of the experimental data**
- **The primary bottleneck is the noise generated by the compute nodes (Tru64)**



Noise reduction: first step

CCS-3

1. removed about 10 daemons from all nodes (including: envmod, insightd, snmpd, lpd, niff)

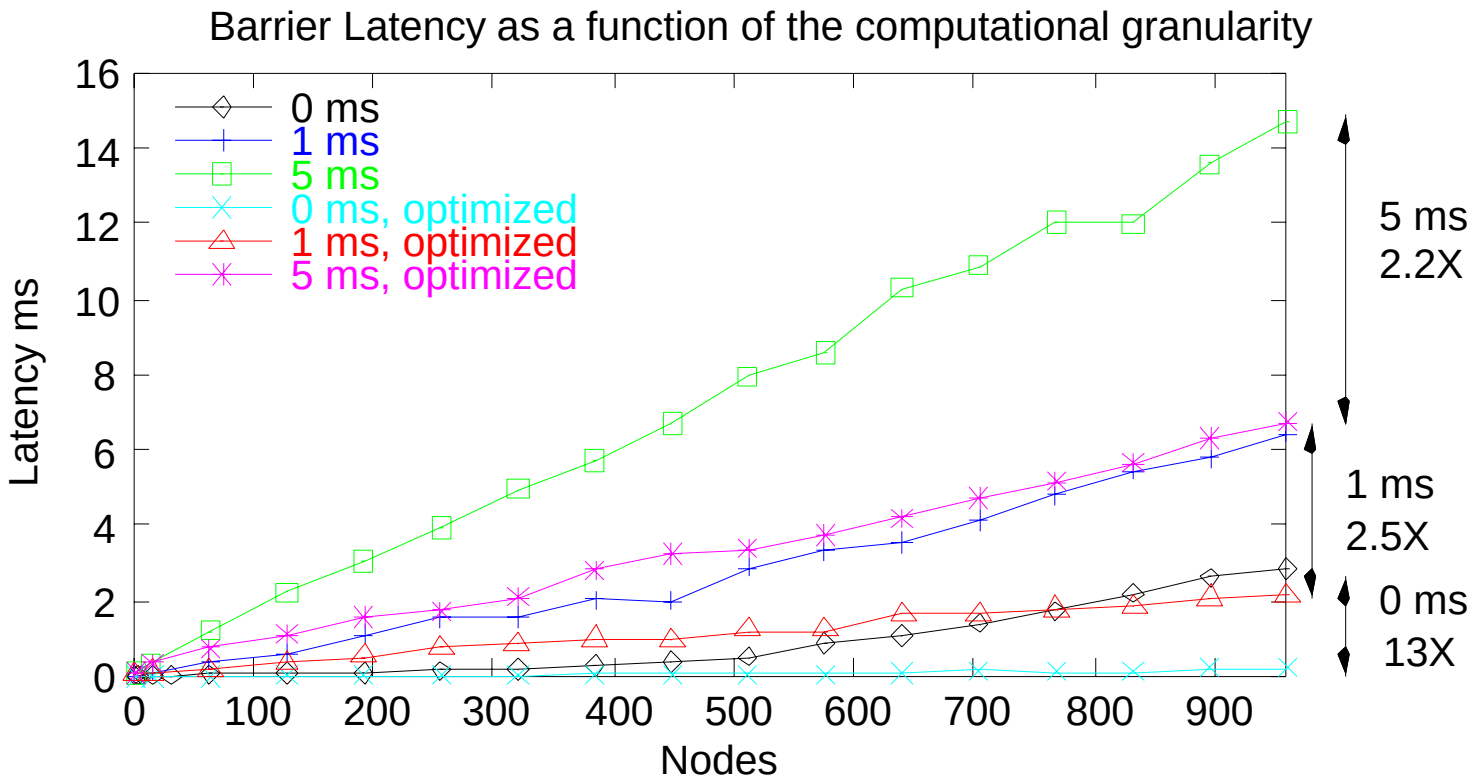
2. decreased the frequency of RMS monitoring by a factor of 2 on each node (from an interval of 30s to 60s)

3. moved several daemons from nodes 1 and 2 to node 0 on each cluster.

Los Alamos

Improvements in the Barrier Synchronization Latency

CCS-3

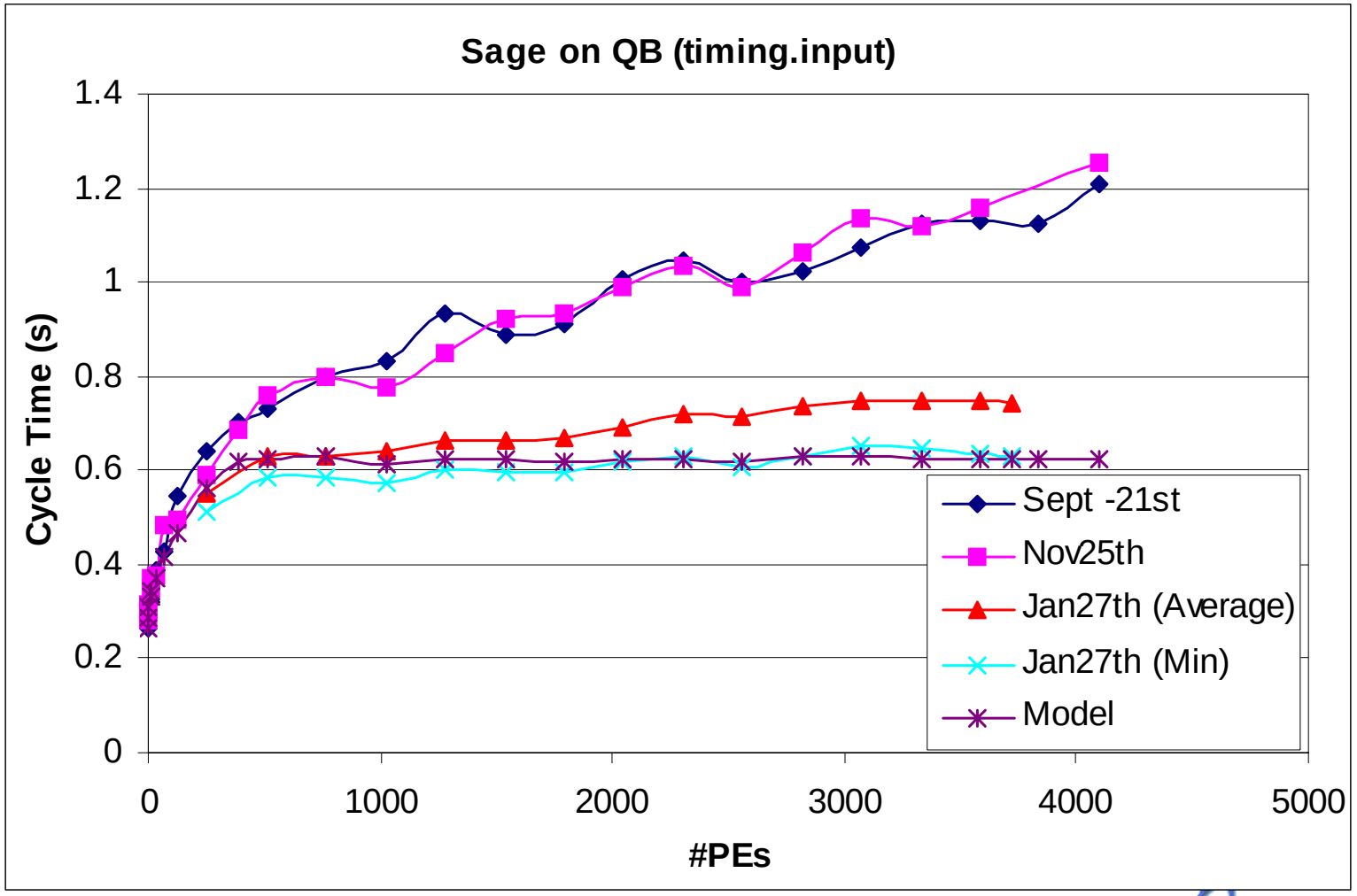


Los Alamos

Noise reduction: second step

CCS-3

- **We configured out nodes 0 and 31**



Los Alamos

Performance Improvements

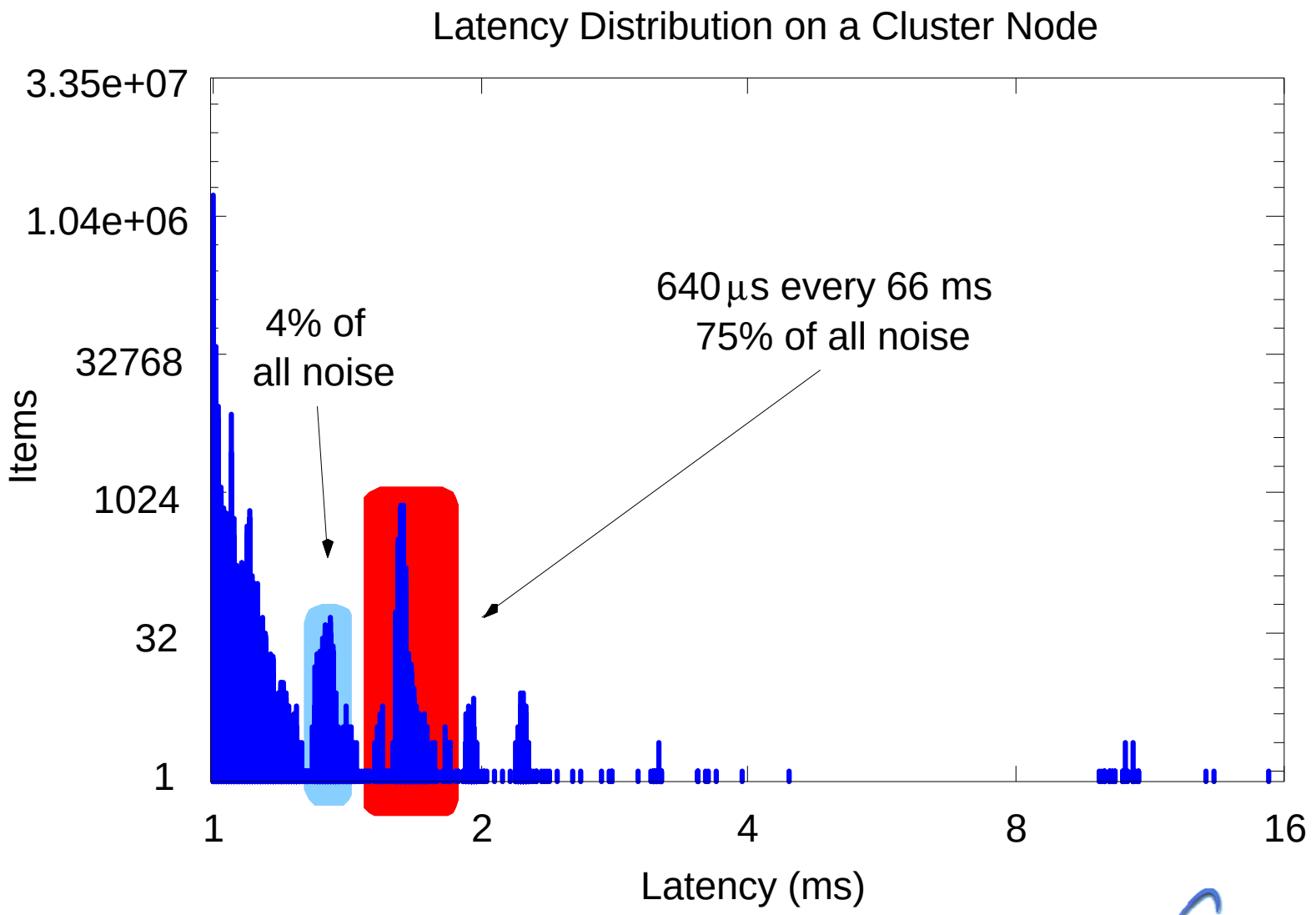
CCS-3

The performance of Sage is up to 55% better on average, and 82% better considering the minimum run time

Number Processors	% Improvement after noise removal	% Potential Total Improvement
512	20%	30%
1024	21%	38%
2048	43%	60%
3072	53%	72%
3716	55%	82%

Los Alamos

Still, a lot of noise left on Q!



CCS-3

Los Alamos

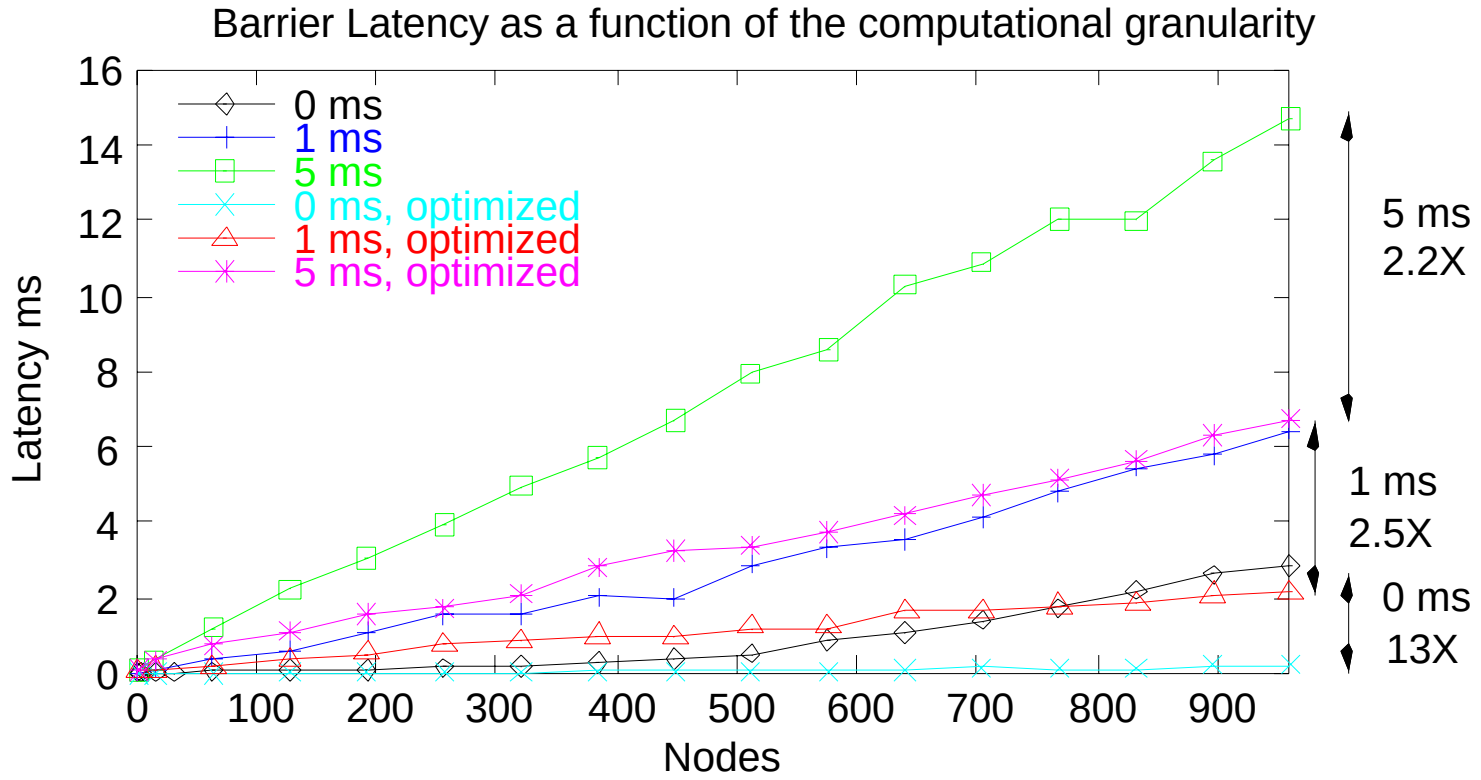
Generalizing these results to other applications other than Sage

- An interesting property correlates the computational granularity of a balanced bulk-synchronous application correlates to the type of noise.
- The intuition is the following: while any source of noise has a negative impact on the overall behavior of the machine, a few sources of noises tend to have a major impact on a given application.
- As a rule of thumb, the computational granularity of the application “enters in resonance” with the noise that has the same magnitude
- The performance can be enhanced by selectively removing sources of noise
- We can provide a reasonable estimate of the performance improvement knowing the computational granularity of a given application.

Impact of Different Sources of Noise

CCS-3

- **We consider the impact of each source of noise for each type of computational granularity in the largest processor configuration**

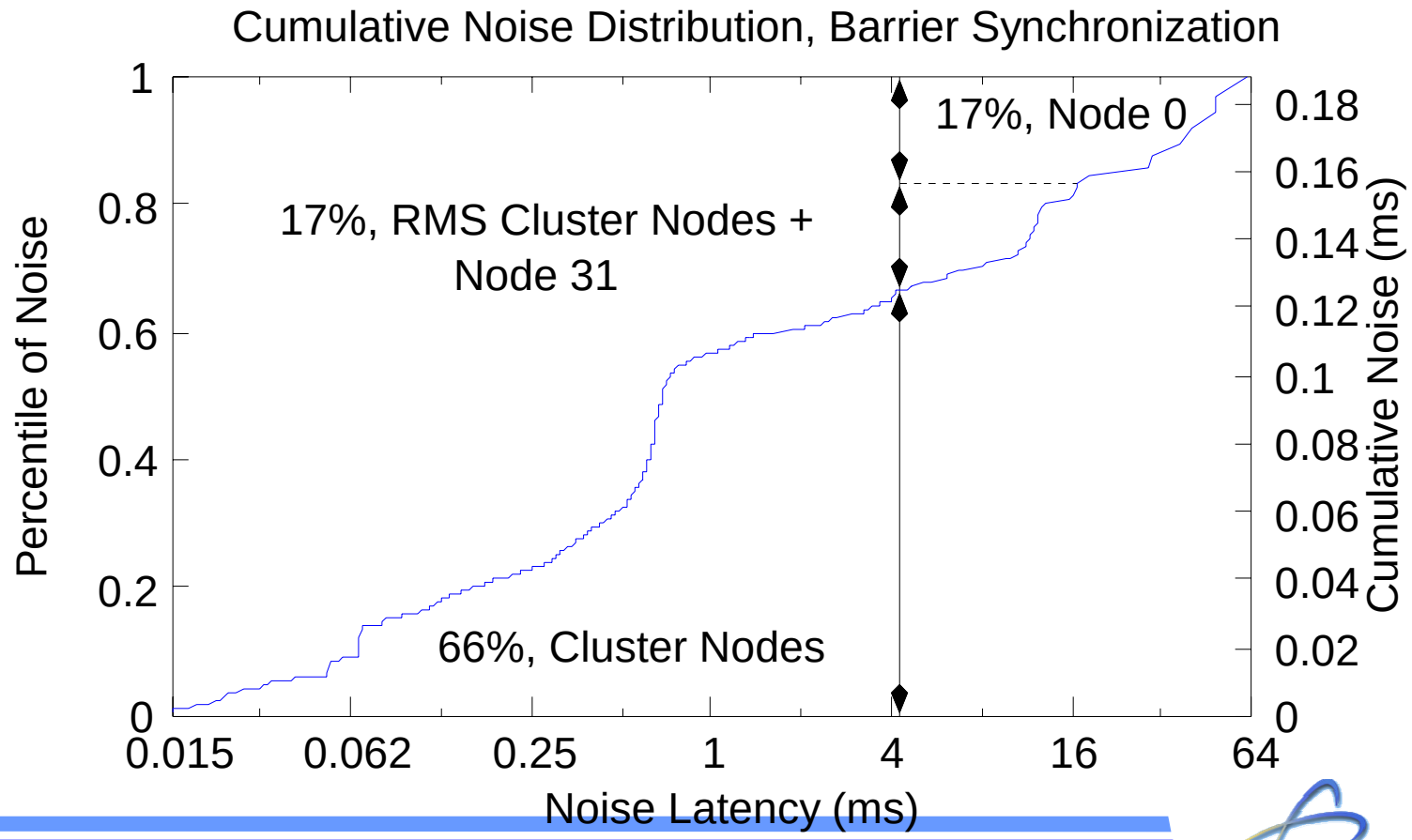


Los Alamos

Cumulative Noise Distribution, Sequence of Barriers with No Computation

CCS-3

- **Most of the latency is generated by the fine-grained, high-frequency noisie of the cluster nodes**

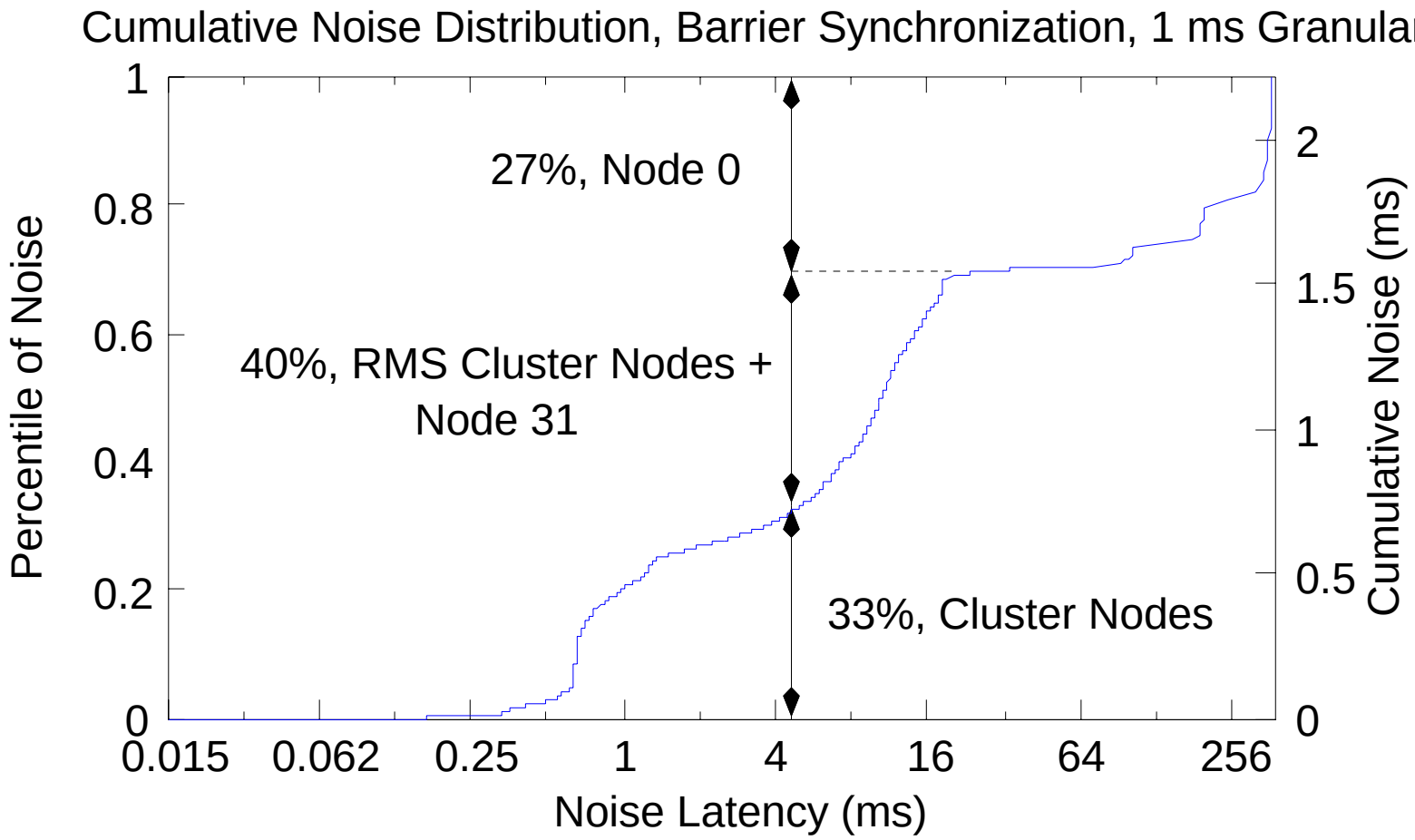


Los Alamos

Barriers with 1 ms of Computational Granularity

CCS-3

- **The major source of noise is now RMS**

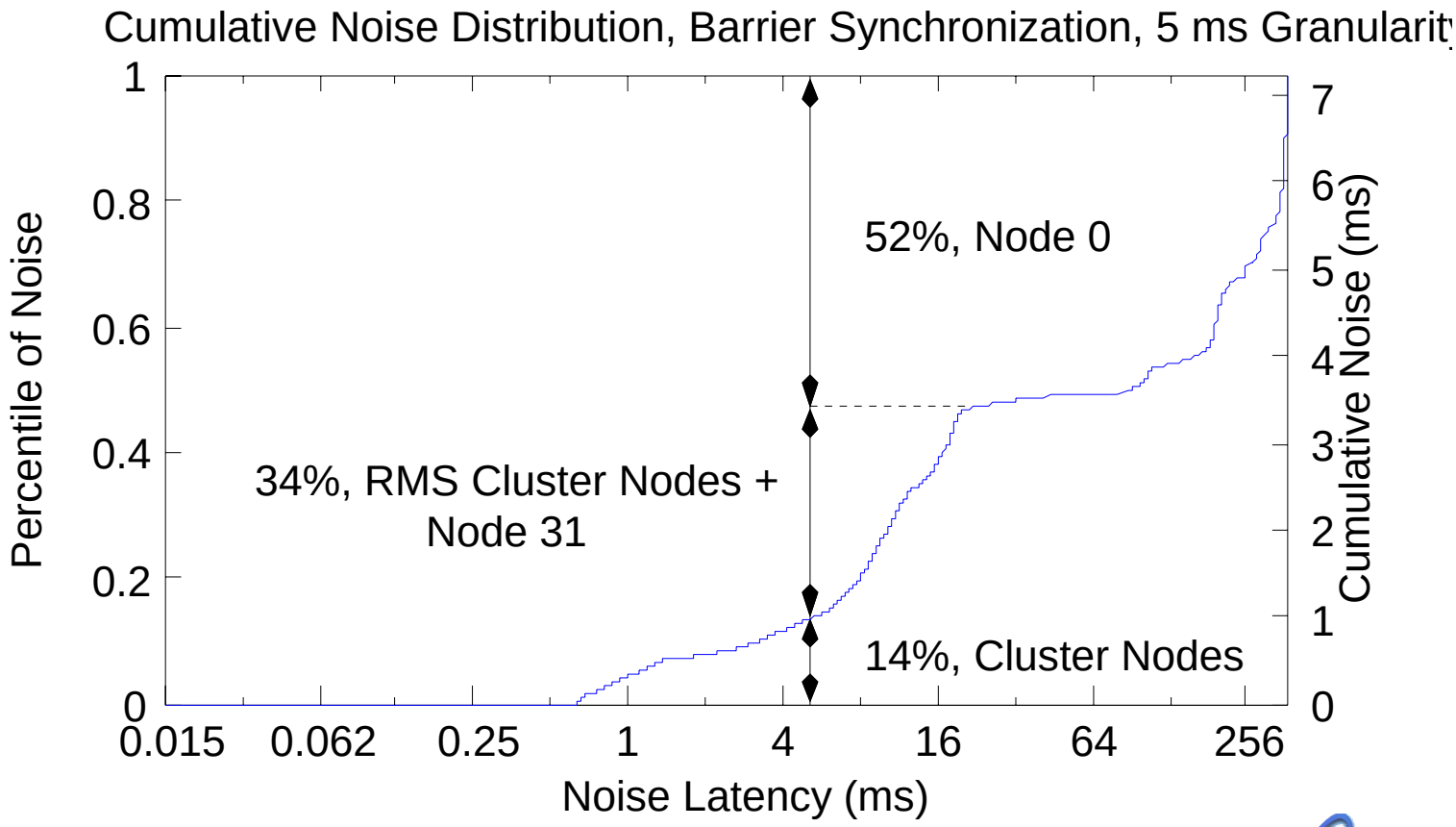


Los Alamos

Barriers with 5ms of Computational Granularity

CCS-3

- **The major source of noise is the low frequency, coarse grained noise of node 0**



Los Alamos

Conclusions

CCS-3

- **Identified performance issues on Q**
  - Used modeling to determine that a problem exists
  - Developed computation kernels to quantify:
    - » frequency and duration of O/S events
  - Effect increases with the # nodes
  - Impact is determined by the computation granularity in an application
- **Successfully explained and fixed the performance issues**
- **Presented a simple methodology to generalize these results to other applications**

Los Alamos

Acknowledgements

CCS-3

- **For access to QB (and for using it in the open):**
  - Manuel Vigil, Ray Miller
- **Support during testing:**
  - Amos Lovato, Joe Kleczka, Malcom Lundin, Rick Light
- **HP / COMPAQ / DIGITAL Marlborough for technical interactions:**
  - Ed Benson, Niraj Srivastava, Dick Foster
- **Quadrics for technical discussions and support on the network**
- **Terry Jones (LLNL) for his insightful comments on the noise elimination**

Report will be available from the PAL homepage:

http://www.c3.lanl.gov/par_arch

Los Alamos